PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-055852
(43)Date of publication of application : 20.02.2002

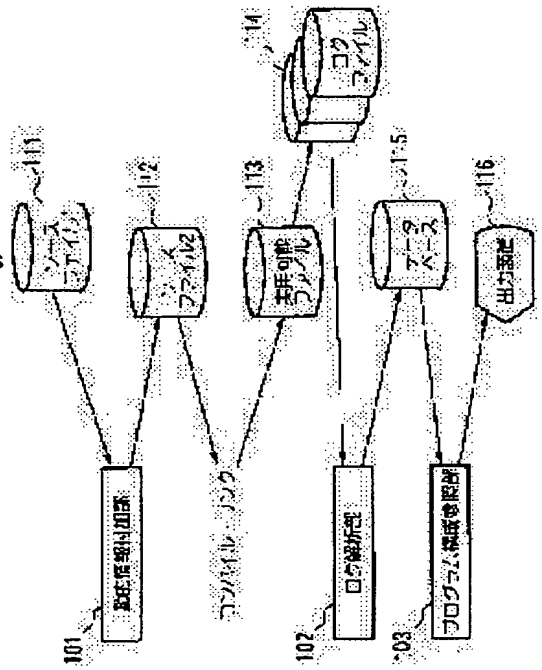
(51)Int.CI. G06F 11/32
G06F 9/44
G06F 11/34

(21)Application number : 2000-239127
(22)Date of filing : 07.08.2000

(71)Applicant : NEC CORP
(72)Inventor : OZAKI YUKIO

(54) OBJECT GENERATION/EXTINCTION INFORMATION MANAGEMENT SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide an object generation/extinction information management system to take out pieces of information about the generation and the extinction of the object when a program is executed and to manage them by registering them in a data base.
SOLUTION: A dynamic information adding part 101 reads a source file 1.111 described in a C++ programming language, embeds a processing so as to output generation and extinction information of the object at timing of the generation and the extinction of the object and generates a source file 2.112. A log analyzing part 102 reads a generated log file 114 by executing an executable file 113 generated from the source file 2.112, analyzes the pieces of generation/extinction information of the object and registers the pieces of information in the data base 115. A program structure referring part 103 takes out the pieces of information registered in the data base 115 and displays them on an output device 116.



LEGAL STATUS

[Date of request for examination] 24.07.2001
[Date of sending the examiner's decision of rejection] 26.10.2004
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－55852
（P2002－55852A）
(43)公開日　平成14年２月20日(2002.2.20)

(51)Int.Cl.'	識別記号	ＦＩ	テーマコード（参考）
Ｇ０６Ｆ　11/32		Ｇ０６Ｆ　11/32	Ａ　５Ｂ０４２
9/44	５３０	9/44	５３０Ｐ
11/34		11/34	Ｌ

審査請求　有　　請求項の数６　ＯＬ　（全７頁）

(21)出願番号　特願2000－239127(P2000－239127)

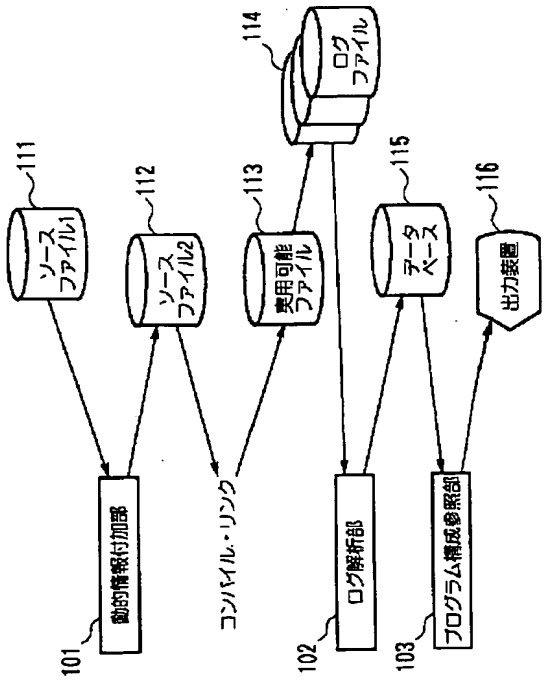
(22)出願日　平成12年８月７日(2000.8.7)

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(72)発明者　尾崎　裕毅男
東京都港区芝五丁目７番１号　日本電気株式会社内
(74)代理人　100108578
弁理士　高橋　詔男　（外３名）

Ｆターム(参考)　5B042 GA02 LA02 MC40 NN07

(54)【発明の名称】　オブジェクトの生成・消滅情報管理方式

(57)【要約】
【課題】　プログラム実行時のオブジェクト生成と消滅の情報を取り出し、データベースに登録して管理するオブジェクトの生成・消滅情報管理方式を提供する。
【解決手段】　動的情報付加部１０１は、Ｃ＋＋プログラミング言語で記述されたソースファイル１・１１１を読み込み、オブジェクトの生成と消滅のタイミングで、オブジェクトの生成と消滅情報を出力するように処理を埋め込み、ソースファイル２・１１２を作成する。ログ解析部１０２は、ソースファイル２・１１２から作成された実行可能ファイル１１３を実行して生成されたログファイル１１４を読み込み、オブジェクトの生成・消滅情報を解析して、データベース１１５に情報を登録する。プログラム構成参照部１０３は、データベース１１５に登録された情報を取り出し、出力装置１１６に表示する。



1

【特許請求の範囲】

【請求項１】　ソースファイルを読みこみ、オブジェクトの生成および消滅について解析し、オブジェクト生成と消滅の情報を出力する処理を埋め込み、編集したソースファイルを生成する動的情報付加部と、

該動的情報付加部により生成されたソースファイルから作成された実行可能ファイルを実行したとき、出力されるオブジェクトの生成および消滅の情報からなるログファイルを解析して解析データを生成するログ解析部と、

該ログ解析部から出力された解析データを格納する記憶部と、

該記憶部に格納されたデータを読み出し、オブジェクトのライフサイクルを表示するデータを出力するプログラム構成参照部と、

を具備することを特徴とするオブジェクトの生成・消滅情報管理方式。

【請求項２】　前記動的情報付加部は、オブジェクト指向プログラム言語により記述されたソースファイルを読み込み、解析することを特徴とする請求項１に記載のオブジェクトの生成・消滅情報管理方式。

【請求項３】　前記動的情報付加部におけるソースファイルのオブジェクトの生成情報出力処理の埋め込みは、オブジェクトが「ｎｅｗ」演算子により生成される場合は、生成後のタイミングで出力するように行い、クラスで使用宣言されている場合は、コンストラクタに追加し、あるいは新しいコンストラクタを追加してその中に追加し、メソッドの中でクラスが使用宣言されている場合は、メソッドの中に追加することを特徴とする請求項１および請求項２に記載のオブジェクトの生成・消滅情報管理方式。

【請求項４】　前記動的情報付加部におけるソースファイルのオブジェクトの消滅情報出力処理の埋め込みは、オブジェクトが「ｄｅｌｅｔｅ」演算子により消滅する場合およびクラス内でクラスの使用宣言しているときに定義もとのクラスが消滅する場合には、クラスのデストラクタに追加し、メソッドの中でクラスが使用宣言されている場合は、メソッドが終了する前に出力するように追加することを特徴とする請求項１ないし請求項３のいずれかに記載のオブジェクトの生成・消滅情報管理方式。

【請求項５】　前記動的情報付加部においてソースファイルに埋め込まれた出力処理によって出力されるオブジェクトの生成および消滅の情報は、オブジェクトの生成および消滅の時間、生成・消滅の区分、生成もととなるオブジェクトの名前、新しく生成されたオブジェクトの名前などから成ることを特徴とする請求項１ないし請求項４のいずれかに記載のオブジェクトの生成・消滅情報管理方式。

【請求項６】　ソースファイルを読み込み、オブジェクトの生成および消滅について解析してオブジェクト生成

2

と消滅の情報を出力する処理を埋め込んだソースファイルを生成し、生成したソースファイルのコンパイルおよびリンクを行い実行可能ファイルを生成する生成部と、

該生成部により生成された実行可能ファイルを実行したとき、出力されるオブジェクトの生成および消滅の情報からなるログファイルを解析して解析データを生成するログ解析部と、

該ログ解析部から出力された解析データを格納する記憶部と、

該記憶部に格納されたデータを読み出し、オブジェクトのライフサイクルを表示するデータを出力するプログラム構成参照部と、

を具備することを特徴とするオブジェクトの生成・消滅情報管理方式。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】この発明は、Ｃ＋＋プログラミング言語で記述されたプログラムの実行時におけるオブジェクトの振る舞いを解析するために、オブジェクトの生成、消滅およびオブジェクト間の関連を情報として取り出し、データベースに登録して管理するオブジェクトの生成・消滅情報管理方式に関する。

【０００２】

【従来の技術】近年、プログラムの開発では、オブジェクト指向プログラミングが主流になり、ＪＡＶＡ（登録商標）、Ｃ＋＋などの言語を用いて多くのアプリケーションプログラムが開発されている。オブジェクト指向によるプログラムは、プログラムを構成する多数のオブジェクトがあり、それらが相互にメッセージを交換することによって処理を行う。また、処理を行う過程でオブジェクトが生成あるいは消滅する。従って、実行時のオブジェクトの振る舞いを解析するために、オブジェクトの生成・消滅のライフサイクルを管理する技術が必要になる。従来、プログラム資産を解析するには、そのオブジェクトの構成関係を出力する方法が用いられていた。

【０００３】

【発明が解決しようとする課題】しかしながら、上述した従来のオブジェクトの構成関係を出力する方法では、実行時に動的に変化するオブジェクトの生成・消滅のライフサイクルを解析し、その使用状態を確認することはできないという問題があった。

【０００４】この発明は、上記の点に鑑みてなされたもので、その目的は、ソースファイルにオブジェクトの生成および消滅の情報を出力する処理を埋め込むことにより、オブジェクトの生成、消滅およびオブジェクト間の関連を情報として取り出し、データベースに登録して管理するオブジェクトの生成・消滅情報管理方式を提供することにある。

【０００５】

【課題を解決するための手段】上記の課題を解決するた

めに、請求項１に記載の発明は、ソースファイルを読みこみ、オブジェクトの生成および消滅について解析し、オブジェクト生成と消滅の情報を出力する処理を埋め込み、編集したソースファイルを生成する動的情報付加部と、該動的情報付加部により生成されたソースファイルから作成された実行可能ファイルを実行したとき、出力されるオブジェクトの生成および消滅の情報からなるログファイルを解析して解析データを生成するログ解析部と、該ログ解析部から出力された解析データを格納する記憶部と、該記憶部に格納されたデータを読み出し、オブジェクトのライフサイクルを表示するデータを出力するプログラム構成参照部とを具備することを特徴とするオブジェクトの生成・消滅情報管理方式である。

【０００６】また、請求項２に記載の発明は、請求項１に記載のオブジェクトの生成・消滅情報管理方式において、前記動的情報付加部は、オブジェクト指向プログラム言語により記述されたソースファイルを読み込み、解析することを特徴とする。

【０００７】また、請求項３に記載の発明は、請求項１および請求項２に記載のオブジェクトの生成・消滅情報管理方式において、前記動的情報付加部におけるソースファイルのオブジェクトの生成情報出力処理の埋め込みは、オブジェクトが「ｎｅｗ」演算子により生成される場合は、生成後のタイミングで出力するように行い、クラスで使用宣言されている場合は、コンストラクタに追加し、あるいは新しいコンストラクタを追加してその中に追加し、メソッドの中でクラスが使用宣言されている場合は、メソッドの中に追加することを特徴とする。

【０００８】また、請求項４に記載の発明は、請求項１ないし請求項３のいずれかに記載のオブジェクトの生成・消滅情報管理方式において、前記動的情報付加部におけるソースファイルのオブジェクトの消滅情報出力処理の埋め込みは、オブジェクトが「ｄｅｌｅｔｅ」演算子により消滅する場合およびクラス内でクラスの使用宣言しているときに定義もとのクラスが消滅する場合には、クラスのデストラクタに追加し、メソッドの中でクラスが使用宣言されている場合は、メソッドが終了する前に出力するように追加することを特徴とする。

【０００９】また、請求項５に記載の発明は、請求項１ないし請求項４のいずれかに記載のオブジェクトの生成・消滅情報管理方式において、前記動的情報付加部においてソースファイルに埋め込まれた出力処理によって出力されるオブジェクトの生成および消滅の情報は、オブジェクトの生成および消滅の時間、生成・消滅の区分、生成もととなるオブジェクトの名前、新しく生成されたオブジェクトの名前などから成ることを特徴とする。

【００１０】また、請求項６に記載の発明は、ソースファイルを読み込み、オブジェクトの生成および消滅について解析してオブジェクト生成と消滅の情報を出力する処理を埋め込んだソースファイルを生成し、生成したソースファイルのコンパイルおよびリンクを行い実行可能ファイルを生成する生成部と、該生成部により生成された実行可能ファイルを実行したとき、出力されるオブジェクトの生成および消滅の情報からなるログファイルを解析して解析データを生成するログ解析部と、該ログ解析部から出力された解析データを格納する記憶部と、該記憶部に格納されたデータを読み出し、オブジェクトのライフサイクルを表示するデータを出力するプログラム構成参照部とを具備することを特徴とするオブジェクトの生成・消滅情報管理方式である。

【００１１】

【発明の実施の形態】（第１の実施例）以下、図面を参照してこの発明の第１の実施例について説明する。図１は、同実施例によるオブジェクトの生成・消滅情報管理方式の構成を示すブロック図である。同図に示すように、本方式は、複数の独立したプログラム制御部を組み合わせてなる。これらプログラム制御部はＣＰＵ（中央演算処理装置）、メモリ、周辺機器から成るハードウェア（図示略）に実装されてその機能が実現される。周辺機器としてキーボード、マウスなどの入力装置、ＣＲＴ（Cathode Ray Tube）や液晶表示装置などの出力装置、外部記憶装置などが接続されるものとする。同図において、１０１は、Ｃ＋＋プログラミング言語で記述された既存資産のソースファイル１・１１１を読み込み、ソースファイル１・１１１内を解析して、オブジェクトの生成と消滅のタイミングで、オブジェクトが生成されたことおよび消滅したことを情報として出力できるように処理を埋め込んでいき、新しくソースファイル２・１１２を作成して出力する動的情報付加部である。同図における矢印はデータの流れを示す。

【００１２】１１３は、ソースファイル２・１１２のコンパイルおよびリンクを行い、作成された実行可能ファイルであり、１１４は、この実行可能ファイル１１３を実行したとき、プログラムの中のオブジェクトが生成、消滅の度に、時間、生成・消滅の区分、生成もととなるオブジェクトの名前、新しく生成されるオブジェクトの名前などを出力したログファイルである。ログファイル１１４の出力場所やログファイル１１４を出力するか否かの指定は、外部から与えられたパラメータによって行われ、動的情報付加部１０１によりソースファイル２・１１２に埋め込み処理される。

【００１３】１０２は、任意のタイミングで、ログファイル１１４を読み込み、ログファイル内のオブジェクトの生成・消滅情報を解析して、データベースへ登録できる形式に変換し、データベース１１５に情報を登録するログ解析部である。１０３は、データベース１１５に登録された情報を取り出し、時系列にそって、オブジェクトのライフサイクルを視覚的に参照し、実行時に動的に変化するオブジェクトのライフサイクルの情報をトレースできるように出力装置１１６に表示するプログラム構

5

成参照部である。プログラム構成参照部１０３は、オブジェクトのライフサイクルの情報を利用しやすいようにファイルへ出力することも可能である。

【００１４】図２は、動的情報付加部１０１の構成を示すブロック図である。同図を参照して動的情報付加部１０１をさらに詳細に説明する。動的情報付加部１０１は、ソースファイル読み込み部２１１、ソースファイル解析部２１２およびソースファイル書き込み部２１３を有するオブジェクトライフサイクル解析処理作成装置２０１から成る。同図における実線の矢印はデータの流れを示し、点線の矢印は制御の流れを示す。ソースファイル読み込み部２１１は、入力装置２２１から入力されたソースファイル１・１１１を読み込み、直接アクセス記憶装置１・２２３に格納する。

【００１５】ソースファイル解析部２１２は、直接アクセス記憶装置１・２２３に格納されている情報を読み込んで、オブジェクトの生成タイミング、消滅タイミングとなる「ｎｅｗ」演算子、「ｄｅｌｅｔｅ」演算子が動作するタイミングや、スタック上に展開されたオブジェクトが解放されるタイミングを探索する。そして、オブジェクトの生成タイミングまたは消滅タイミングが見つかれば、その時点の時間と生成、消滅の区分、生成もとなるオブジェクトの名前、新しく生成されるオブジェクトの名前などを実行時にログとして出力できるように、処理を埋め込み、埋め込んだソースファイルを直接アクセス記憶装置２・２２４に書き込む。ソースファイル書き込み部２１３は、直接アクセス記憶装置２・２２４からソースファイルを読み込み、新しいソースファイル２・１１５として出力する。

【００１６】図３は、ログ解析部１０２の構成を示すブロック図である。同図を参照してログ解析部１０２をさらに詳細に説明する。ここで、実行可能ファイル１１３は、実行可能ファイル実行部３０１により、実行され、オブジェクトの生成および消滅のタイミングで情報を出力し、ログファイル１１４が生成されているものとする。ログファイル読み込み部３０２は、ログファイル１１４を読み込み、読み込んだ内容を記憶装置１・３１３に格納する。ログ解析部３０３は、記憶装置１・３１３の内容を読み込んで解析を行う。記憶装置１・３１３から読み込まれた内容はデータベースへ登録することができる形式に変換されて、記憶装置２・３１４に格納される。データベース登録部３０４は、記憶装置２・３１４に格納されている情報を読み出し、データベース１１５に登録する。

【００１７】図５は、動的情報付加部１０１の処理の流れを示す図であり、図６は、ログ解析部１０２の処理の流れを示す図である。以下、図５および図６を参照して本実施例の動作を説明する。先ず、図５に示すステップＳ５１１で、ソース読み込み処理が動作し、動的情報付加部１０１にソースファイル１・１１１が読み込まれ

6

（ステップＳ５２１）、メモリ上に展開される。次に、ステップＳ５２２では、バックアップ処理により、オリジナルのソースファイルであるソースファイル１・１１１が、上書きされないようにバックアップされる。

【００１８】次いで、ソース解析処理（ステップＳ５１２）が動作し、ステップＳ５２３に進む。ステップＳ５２３では、ステップＳ５２１でメモリ上に展開されたソースファイル１・１１１を初めから順に探索してオブジェクトの生成またはオブジェクトの消滅が発生する箇所を探す。そして、オブジェクトの生成または消滅が発生する箇所が見つかれば、動的情報の埋め込み処理によって時間、オブジェクトの生成と消滅の区分、生成もとなるオブジェクトの名前、新しく生成されるオブジェクトの名前などの情報をログとして出力する処理を埋め込んでいく（ステップＳ５２４）。

【００１９】ここで、出力処理の埋め込みは、次の手順で行われる。Ｃ＋＋プログラミング言語では、オブジェクトの生成は、「ｎｅｗ」演算子を用いた場合、クラスやメソッド内で使用宣言された場合などが考えられる。「ｎｅｗ」演算子で生成される場合は、生成後のタイミングでログ出力するように処理を追加する。クラスで使用宣言された場合は、コンストラクタにログ出力する処理を追加する。コンストラクタが定義されている場合は、既存のコンストラクタに処理を追加し、コンストラクタが定義されていない場合は、新たにコンストラクタを追加して、その中にログ出力する処理を追加する。メソッドの中でクラスが使用宣言された場合は、メソッドの内部にログ出力する処理を追加する。

【００２０】Ｃ＋＋プログラミング言語では、オブジェクトの消滅は、「ｄｅｌｅｔｅ」演算子を用いた場合、クラス内でクラスの使用宣言をしていたときに定義元のクラスが消滅する場合、メソッド内でクラスの使用宣言をしていたときにメソッドが終了する場合などが考えられる。「ｄｅｌｅｔｅ」演算子でオブジェクトが消滅する場合は、クラスのデストラクタにログ出力するように処理を追加する。クラス内でクラスの使用宣言をしていたときに定義もとのクラスが消滅するときも、クラスのデストラクタにログ出力するように処理を追加する。メソッド内でクラスの使用宣言をしていたときには、メソッドが終了する前にログ出力するように処理を追加する。

【００２１】次に、ステップＳ５１３に移り、上述の処理により生成されたソースファイル２・１１２の出力先が決定され、出力処理により出力される（ステップＳ５２５）。動的情報付加部１０１から出力されたソースファイル２・１１２に対してコンパイルおよびリンクが行われ、実行可能ファィル１１３が作成される。次いで、実行可能ファィル１１３が実行され、実行時に、オブジェクトの生成または消滅時のタイミングでオブジェクトのライフサイクル情報をファイルに出力し、ログファイ

7

ル１１４が生成される。次に、図６を参照してログ解析部１０２におけるログファイル１１４の解析および登録の処理を説明する。先ず、オブジェクトライフサイクル情報登録処理（ステップＳ６０１）では、ログファイル１１４の存在する場所を特定してトランザクション制御処理（ステップＳ６１１）を呼び出す。

【００２２】トランザクション制御処理（ステップＳ６１１）では、１つのログファイルの情報登録を１トランザクションとする。次に、ログファイル読み込み処理（ステップＳ６２１）では、指定された場所に格納されているログファイル１１４を読み込み、メモリ上に展開する。そして、メモリ上に展開された内容を元にログ内容の解析処理を行い、生成したデータをデータベースに登録できる形式に変換していく（ステップＳ６２２）。ステップＳ６２３により、この変換されたデータは、データベース１１５へ登録される。登録に失敗した場合は、トランザクションはロールバックされる。トランザクション制御処理（ステップＳ６１１）は、続いて読み込み処理を行うログファイルがある場合は、順次読みこませ、上述の処理を繰り返す。次いで、プログラム構成参照部１０３は、データベース１１５に格納された情報を読み出し、オブジェクトの生成・消滅のライフサイクルを時系列にそって参照できるように出力装置１１６に表示させる。

【００２３】（第２の実施例）以下、本発明の第２の実施例について図面を参照して説明する。図４は、同実施例によるオブジェクトの生成・消滅情報管理方式の構成を示すブロック図である。なお、同図において、第１の実施例の図１と同一部分には同一符号を付してその説明を省略する。図４において、４００は、プリプロセッサ部４０１と、コンパイル・リンク４０２を有し、Ｃ＋＋プログラミング言語のソースファイル４１１を読みこみ、オブジェクトの生成・消滅情報の出力処理を追加し、自動的にコンパイルとリンクを行い実行可能ファイル４１３を生成する生成部である。

【００２４】次に、以上のように構成した実施例の動作を説明する。先ず、プリプロセッサ部４０１は、Ｃ＋＋プログラミング言語で記述された既存資産のソースファイル４１１を読み込む。そして、プログラム構造を解析して、オブジェクトの生成または消滅のタイミングで、オブジェクトが生成されたこと、消滅したことを情報として出力できるように処理を埋め込む。この処理を埋め込んで新しく作成されたソースファイルは、新ソースファイル４１２として記憶装置に出力される。コンパイル・リンク４０２は、新ソースファイル４１２に対して、コンパイルおよびリンクを行い、実行可能ファイル４１３を作成する。

【００２５】実行可能ファイル４１３を実行したとき、プログラムの中のオブジェクトは生成または消滅の度に

8

時間、生成・消滅の区分、生成もととなるオブジェクトの名前、新しく生成されるオブジェクトの名前などをログファイル４１４に出力する。ログ解析部１０２は、ログファイル４１４の情報を読みこみ、ログ情報を解析して、データベースへ登録できる形式に変換し、データベース１１５に情報を登録する。プログラム構成参照部１０３は、データベース１１５に登録された情報を、読み出し、出力装置１１６に表示する。

【００２６】

【発明の効果】以上説明したように、本発明によれば、ソースファイルを解析してオブジェクトの生成・消滅情報を出力する処理を埋め込み、プログラムの実行時に出力するオブジェクトの生成・消滅情報をデータベースに登録して管理するので、プログラム実行時にしか分からない、動的に変化するオブジェクトの生成・消滅のライフサイクルを解析することができる。また、時系列におけるオブジェクトの生成・消滅状態、解放漏れ、オブジェクト間の構成(オブジェクトの親子関係)などを解析することが可能になり、実行時のオブジェクトの構成を容易に理解できるようになる。

【図面の簡単な説明】

【図１】　この発明の第１の実施例の構成を示すブロック図である。

【図２】　同実施例の動的情報付加部の構成を示すブロック図である。

【図３】　同実施例のログ解析部の構成を示すブロック図である。

【図４】　この発明の第２の実施例の構成を示すブロック図である。

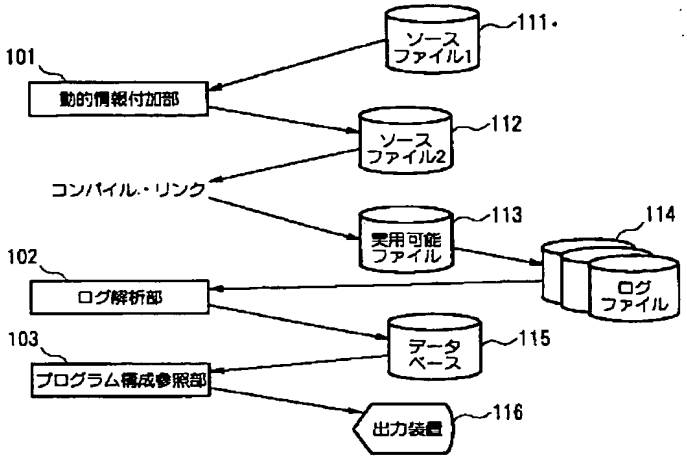
【図５】　第１の実施例の動的情報付加部の処理の流れを示す図である。

【図６】　同実施例のログ解析部の処理の流れを示す図である。

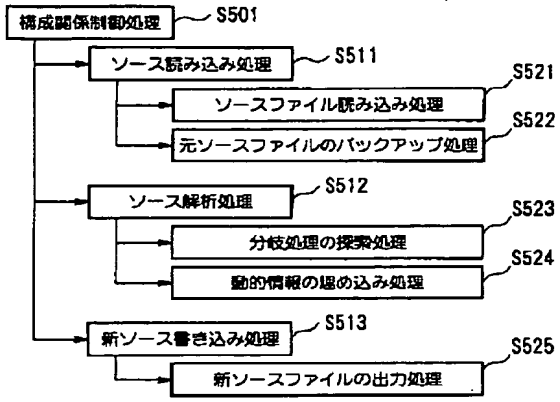
【符号の説明】

１０１　動的情報付加部
１０２　ログ解析部
１０３　プログラム構成参照部
１１５　データベース
１１６　出力装置
２０１　オブジェクトライフサイクル解析処理作成装置
２１１　ソースファイル読み込み部
２１２　ソースファイル解析部
２１３　ソースファイル書き込み部
３０２　ログファイル読み込み部
３０３　ログ解析部
３０４　データベース登録部
４００　生成部
４０１　プリプロセッサ部
４０２　コンパイル・リンク

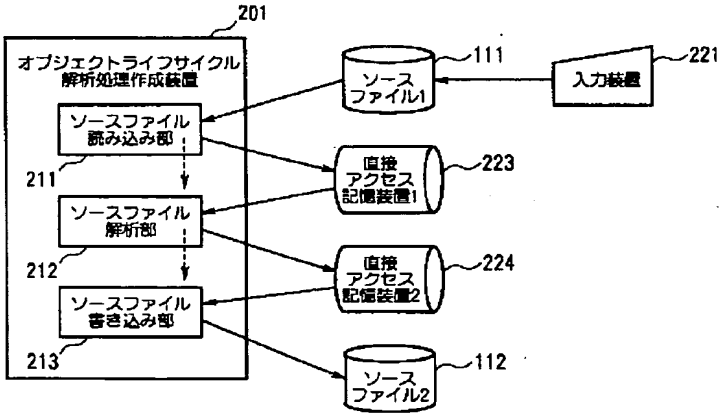
【図１】


ソースファイル1
111
101
動的情報付加部
ソースファイル2
112
コンパイル・リンク
実用可能ファイル
113
114
ログファイル
102
ログ解析部
データベース
115
103
プログラム構成参照部
出力装置
116

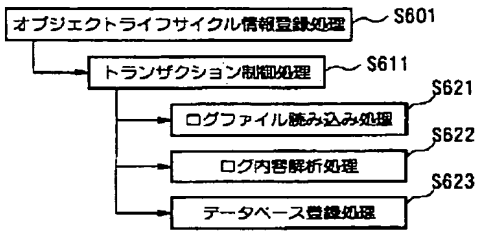
【図５】


構成関係制御処理
S501
ソース読み込み処理
S511
S521
ソースファイル読み込み処理
S522
元ソースファイルのバックアップ処理
ソース解析処理
S512
S523
分岐処理の探索処理
S524
動的情報の埋め込み処理
新ソース書き込み処理
S513
S525
新ソースファイルの出力処理

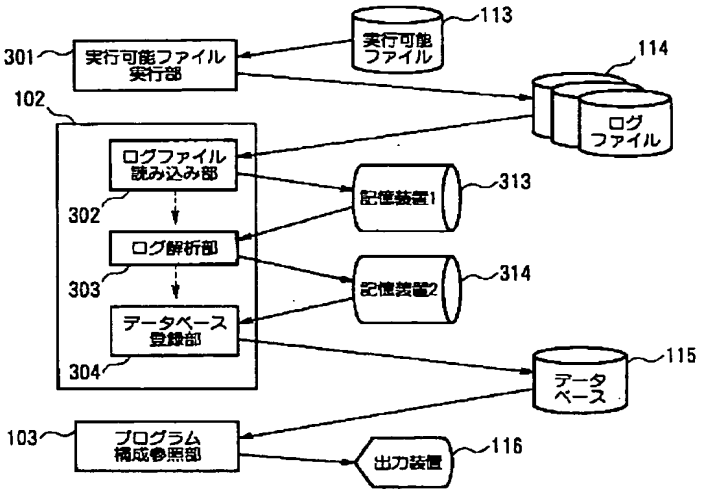
【図２】


201
オブジェクトライフサイクル解析処理作成装置
ソースファイル読み込み部
211
ソースファイル解析部
212
ソースファイル書き込み部
213
ソースファイル1
111
入力装置
221
直接アクセス記憶装置1
223
直接アクセス記憶装置2
224
ソースファイル2
112

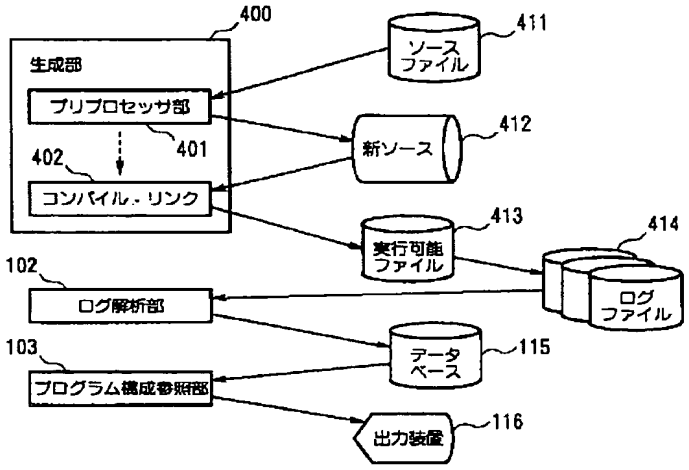
【図６】


オブジェクトライフサイクル情報登録処理
S601
トランザクション制御処理
S611
S621
ログファイル読み込み処理
S622
ログ内容解析処理
S623
データベース登録処理

【図３】


113
実行可能ファイル
301
実行可能ファイル実行部
114
ログファイル
102
ログファイル読み込み部
313
記憶装置1
302
ログ解析部
303
314
記憶装置2
データベース登録部
304
115
データベース
103
プログラム構成参照部
116
出力装置

【図４】


400
411
生成部
ソースファイル
プリプロセッサ部
401
412
新ソース
402
コンパイル・リンク
413
414
実行可能ファイル
ログファイル
102
ログ解析部
103
115
データベース
プログラム構成参照部
116
出力装置